AMENDMENTS TO THE CLAIMS

Claim 1 (Currently Amended): A method ~~of~~ for encrypting including calculating a result E of an exponentiation B^d, B being a base and d being an exponent, wherein the exponent can be described by a binary number from a plurality of bits, comprising the following steps:

initializing a first auxiliary quantity X to a value of 1;

initializing a second auxiliary quantity Y to the base B;

sequentially processing the bits of the exponent by:

first updating the first auxiliary quantity X by X^2 or by a value derived from X^2 and updating the second auxiliary quantity Y by X*Y or by a value derived from X*Y, if a bit of the exponent equals 0, or

second updating the first auxiliary quantity X by X*Y or by a value derived from X*Y and updating the second auxiliary quantity Y by Y^2 or by a value derived from Y^2, if a bit of the exponent equals 1; and

after sequentially processing all the bits of the exponent, using the value of the first auxiliary quantity X as the result of the exponentiation in order to secure data transmission.

Claim 2 (Original): The method according to claim 1, wherein in the step of sequentially processing is started from the most significant bit of the exponent.

Claim 3 (Original): The method according to claim 1,

wherein the exponentiation is a modular exponentiation B^d mod N, N being the module, and

wherein the value derived from X^2, XY or Y^2 is generated by a modular reduction with the module N of X^2, XY and Y^2, respectively.

Claim 4 (Currently Amended): The method according to claim 1,

wherein in the second step of updating, if the bit of the exponent equals 1, the value ~~X^2~~-Y^2 and the value X*Y are calculated parallel to each other.

Claim 5 (Currently Amended): The method according to claim 1,

wherein in the first step of updating, if the bit equals 0, the value X*Y and the value ~~Y^2~~-X^2 are calculated parallel to each other.

Claim 6 (Original): The method according to claim 3,

wherein the modular exponentiation is used in an RSA decryption and/or an RSA encryption.

Claim 7 (Original): The method according to claim 3,

wherein the exponent d, the base B and/or the module N are integers.

Claim 8 (Original): A device for calculating a result E of an exponentiation B^d, B being a base and d being an exponent, wherein the exponent can be described by a binary number from a plurality of bits, comprising:

an initializer for initializing a first auxiliary quantity X to a value of 1 and a second auxiliary quantity Y to the base B; and

a processor for sequentially processing the bits of the exponent by:

> updating the first auxiliary quantity X by X^2 or by a value derived from X^2 and updating the second auxiliary quantity Y by X*Y or by a value derived from X*Y, if a bit of the exponent equals 0, or

updating the first auxiliary quantity X by X*Y or by a value derived from X*Y and updating the second auxiliary quantity Y by Y^2 or by a value derived from Y^2, if a bit of the exponent equals 1;

wherein the processor is operative to use the value of the first auxiliary quantity X as the result of the exponentiation after having sequentially processed all the bits of the exponent.

Claim 9 (Original): The device according to claim 8,

wherein the processor for sequentially processing comprises a first calculating unit and a second calculating unit, the first calculating unit and the second calculating unit being arranged to operate parallel to each other, and

wherein the first calculating unit is arranged to calculate X^2 if the bit of the exponent equals 0 or to calculate X*Y if the bit of the exponent equals 1, and

wherein the second calculating unit is arranged to calculate X*Y if the bit equals 0 and to calculate Y^2 if the bit equals 1.